TD d'algorithmique avancée

Corrigé du TD 3 : multiplications « diviser pour régner »

Jean-Michel Dischler et Frédéric Vivien

Multiplications « diviser pour régner »

1. Montrez comment multiplier deux polynômes linéaires $ax+b$ et $cx+d$ à l'aide de trois multiplications seulement. (*Indication* : l'une des multiplications est $(a+b)(c+d)$.)
   D'une part : $(ax+b)\times(cx+d) = acx^2+(ad+bc)x+bd$. *D'autre part :* $(a+b)(c+d) = ac+ad+bc+bd = ac+bd+ad+bc$. *D'où :* $(ax+b)\times(cx+d) = acx^2+((a+b)(c+d)-ac-bd)x+bd$, *et les trois seules multiplications nécessaires sont les calculs : ac, bd et* $(a+b)(c+d)$.
2. Donnez deux algorithmes « diviser pour régner » permettant de multiplier deux polynômes de degré au plus n et s'exécutant en $\Theta(n^{\log_2 3})$.
   (a) Le premier algorithme devra couper les coefficients du polynôme d'entrée en deux moitiés, l'une supérieure et l'autre inférieure.
   Soient $P[X]$ *et* $Q[X]$ *les deux polynômes d'entrée.* $P[X] = \sum_{i=0}^{n} p_i X^i$ *et* $Q[X] = \sum_{i=0}^{n} q_i X^i$.

$$\begin{aligned} P[X] &= \sum_{i=0}^{n} p_i X^i \\ &= \left(\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} p_i X^i\right) + \left(\sum_{i=1+\lfloor \frac{n}{2} \rfloor}^{n} p_i X^i\right) \\ &= \left(\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} p_i X^i\right) + \left(X^{1+\lfloor \frac{n}{2} \rfloor} \sum_{i=0}^{n-1-\lfloor \frac{n}{2} \rfloor} p_{i+1+\lfloor \frac{n}{2} \rfloor} X^i\right). \end{aligned}$$

   On pose alors $B[X] = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} p_i X^i$ *et* $A[X] = \sum_{i=0}^{n-1-\lfloor \frac{n}{2} \rfloor} p_{i+1+\lfloor \frac{n}{2} \rfloor} X^i$. $A[X]$ *et* $B[X]$ *sont alors deux polynômes de degré au plus* $\lfloor \frac{n}{2} \rfloor$ *et* $P[X] = A[X]X^{1+\lfloor \frac{n}{2} \rfloor} + B[X]$. *On définit de même les polynômes* C *et* D *pour* Q *:* $Q[X] = C[X]X^{1+\lfloor \frac{n}{2} \rfloor} + D[X]$.
   Avec les notations précédemment définies, on a :

$$\begin{aligned} P[X]Q[X] = \; & A[X]C[X]X^{2+2\lfloor \frac{n}{2} \rfloor} \\ & +((A[X]+B[X])(C[X]+D[X]) - A[X]C[X] - B[X]D[X])X^{1+\lfloor \frac{n}{2} \rfloor} \\ & +C[X]D[X]. \end{aligned}$$

   Par conséquent, le produit de deux polynômes de degré au plus n peut se ramener au calcul de trois produits de polynômes de degré au plus $\lfloor \frac{n}{2} \rfloor$ *(*$A[X]C[X]$, $B[X]D[X]$ *et* $(A[X]+B[X])(C[X]+D[X])$*), à des additions de polynômes de degré au plus n —ce qui coûte* $\Theta(n)$*— et à des multiplications par un monôme* X^j *—ce qui est un simple décalage des indices et coûte également* $\Theta(n)$. *L'équation de récurrence définissant la complexité de notre algorithme est alors :*

$$T(n) = 3T\left(\frac{n}{2}\right) + \Theta(n).$$

   Nous appliquons alors le théorème vu en cours :

Théorème 1 (Résolution des récurrences « diviser pour régner »).
Soient $a \geq 1$ et $b > 1$ deux constantes, soit $f(n)$ une fonction et soit $T(n)$ une fonction définie pour les entiers positifs par la récurrence :

$$T(n) = aT(n/b) + f(n),$$

où l'on interprète n/b soit comme $\lfloor n/b \rfloor$, soit comme $\lceil n/b \rceil$.
$T(n)$ peut alors être bornée asymptotiquement comme suit :

i. Si $f(n) = O(n^{(\log_b a)-\epsilon})$ pour une certaine constante $\epsilon > 0$, alors $T(n) = \Theta(n^{\log_b a})$.
ii. Si $f(n) = \Theta(n^{\log_b a})$, alors $T(n) = \Theta(n^{\log_b a} \log n)$.
iii. Si $f(n) = \Omega(n^{(\log_b a)+\epsilon})$ pour une certaine constante $\epsilon > 0$, et si $af(n/b) \leq cf(n)$ pour une constante $c < 1$ et n suffisamment grand, alors $T(n) = \Theta(f(n))$.

Ici $a = 3$, $b = 2$ et $f(n) = \Theta(n)$. Comme $\log_2 3 > 1$, nous nous trouvons dans le cas i) du théorème et donc

$$T(n) = \Theta\left(n^{\log_2 3}\right).$$

Pour fixer les idées, $\log_2 3 \approx 1,58$ et l'algorithme naïf de multiplications de polynômes est en $\Theta(n^2)$.

(b) Le second algorithme devra séparer les coefficients du polynôme d'entrée selon la parité de leur indice.

$$\begin{aligned} P[X] &= \sum_{i=0}^{n} p_i X^i \\ &= \sum_{i \ pair} p_i X^i + \sum_{i \ impair} p_i X^i \\ &= \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} p_{2i} X^{2i} + \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} p_{2i+1} X^{2i+1} \\ &= \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} p_{2i} X^{2i} + X \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} p_{2i+1} X^{2i} \end{aligned}$$

On pose alors $A[X] = \sum_{i=1}^{\lfloor \frac{n-1}{2} \rfloor} p_{2i+1} X^i$ et $B[X] = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} p_{2i} X^i$. $A[X]$ et $B[X]$ sont alors deux polynômes de degré au plus $\lfloor \frac{n}{2} \rfloor$ et $P[X] = A[X^2]X + B[X^2]$. On définit de même les polynômes C et D pour Q : $Q[X] = C[X^2]X + D[X^2]$.
Avec les notations précédemment définies on a :

$$\begin{aligned} P[X]Q[X] &= A[X^2]C[X^2]X^2 \\ &+((A[X^2] + B[X^2])(C[X^2] + D[X^2]) - A[X^2]C[X^2] - B[X^2]D[X^2])X \\ &+B[X^2]D[X^2] \end{aligned}$$

Par conséquent, le produit de deux polynômes de degré au plus n peut se ramener au calcul de trois produits de polynômes de degré au plus $\lfloor \frac{n}{2} \rfloor$ ($A[X]C[X]$, $B[X]D[X]$ et $(A[X]+B[X])(C[X]+D[X])$), à des additions de polynômes de degré au plus n —ce qui coûte $\Theta(n)$— à des multiplications par un monôme X^j —ce qui est un simple décalage des indices et coûte également $\Theta(n)$— et à des transpositions du polynôme $R[X]$ au polynôme $R[X^2]$ —ce qui est encore un simple décalage des indices et coûte également $\Theta(n)$. L'équation de récurrence définissant la complexité de notre algorithme est donc comme précédemment :

$$T(n) = 3T\left(\frac{n}{2}\right) + \Theta(n),$$

et la complexité est la même :

$$T(n) = \Theta\left(n^{\log_2 3}\right).$$

3. Montrez que deux entiers à n bits peuvent être multipliés en $\Theta(n^{\log_2 3})$ étapes.
   L'entier $m = \sum_{i=0} m_i 2^i$ peut être vu comme un polynôme : il nous suffit de réappliquer un des algorithmes vu à la question précédente pour obtenir le résultat escompté.

Calcul de (cos(nx), sin(nx))

Écrire un algorithme prenant en entrée un entier `n` et une paire de valeurs réelles qui sont en fait les valeurs du cosinus et du sinus d'un certain angle `x`, et renvoyant la paire `(cos(nx), sin(nx))`. Autrement dit, le deuxième argument de la fonction est une paire `(a,b)` telle que `a = cos x` et `b = sin x`. Le schéma de calcul doit être récursif (mais non « diviser pour régner »).

On pourra se servir des formules de trigonométrie suivantes :

```
cos(nx) = cos((n-1)x) cos(x) - sin((n-1)x) sin(x)
sin(nx) = sin((n-1)x) cos(x) + cos((n-1)x) sin(x)
```

TRIGO(n, a, b)
  Si $n = 0$ **alors renvoyer** $(1, 0)$
    sinon $c, d =$ TRIGO$(n-1, a, b)$
      renvoyer $(ac - bd, ad + bc)$